(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 December 2002 (27.12.2002)

PCT

(10) International Publication Number
WO 02/103536 A1

(51) International Patent Classification[7]: G06F 15/00

(21) International Application Number: PCT/KR02/01157

(22) International Filing Date: 19 June 2002 (19.06.2002)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:
2001/34583 19 June 2001 (19.06.2001) KR
2001/50151 21 August 2001 (21.08.2001) KR

(71) Applicant *(for all designated States except US)*: **TERUTEN INC.** [KR/KR]; 3F, Seongwhan bldg, 901-66 Daechi-dong, Gangnam-gu, Seoul 135-841 (KR).

(72) Inventors; and

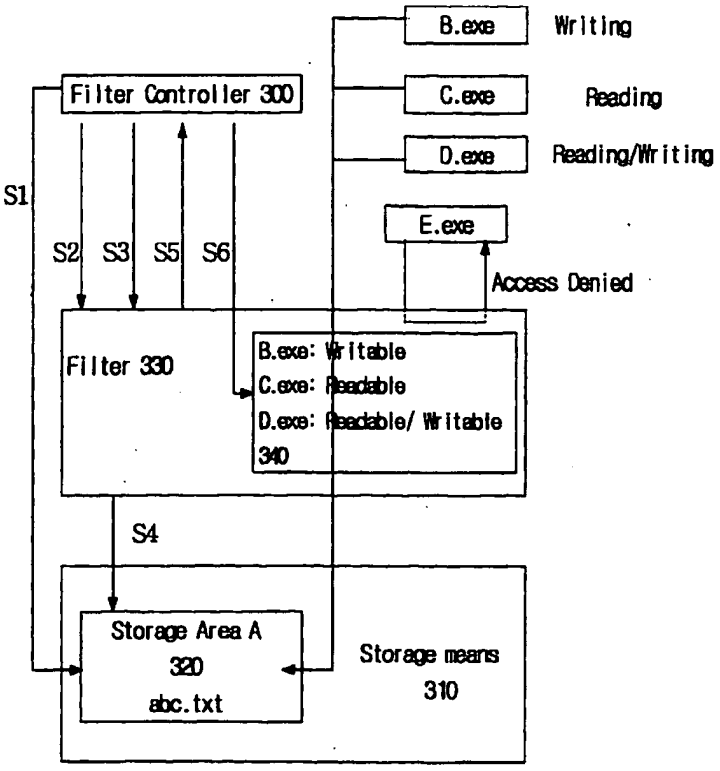
(75) Inventors/Applicants *(for US only)*: **YOON, Seokgu** [KR/KR]; 505-1703 WooSung APT, BumGye-dong, DongAn-gu, AnYang 431-724 (KR). **KIM, Sungyup** [KR/KR]; 372-2 GunJa-dong, GwangJin-gu, Seoul 143-840 (KR). **LEE, Saerock** [KR/KR]; 202-1208 WooSung 2cha APT., JaYang3-dong, GwangJin-gu, Seoul 143-193 (KR). **LEE, Young** [KR/KR]; 505-1703 WooSung APT., BumGye-dong, DongAn-gu, AnYang 431-724 (KR).

(74) Agent: **PARK, Hyuncheol**; 196-180 Bongcheon11-dong, Gwanak-gu, Seoul 151-817 (KR).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

*[Continued on next page]*

(54) Title: SECURITY SYSTEM FOR SECURING EXCHANGE AND EXECUTION OF DIGITAL DATA




(57) Abstract: Disclosed are a method and a system for protecting digital data that are capable of confining a predetermined authenticated execution program only to input/output and execute digital data. The system according to the invention includes first storage means for receiving and storing data, access control means for storing identification information about a data execution program, determination means connected to the access control means for determining whether or not the identification information about a data-requesting execution program has been recorded in the access control means if the data-requesting execution program requests for a data file, and transmission means connected to the first storage means and the determination means for transmitting the data to the data-requesting execution program from the first storage means so as to be executed upon receipt of a determination signal that the identification information about the execution program has been recorded from the determination means.

WO 02/103536 A1

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**
— *with international search report*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

# SECURITY SYSTEM FOR FOR SECURING EXCHANGE AND EXECUTION OF DIGITAL DATA

## Technical Field

The present invention is related to a computer system using digital data supplied on-line or off-line and a method as well as a program medium therefore, and in particular, a digital data protection technology for preventing duplication, distribution or use of digital data by unauthorized users while providing user convenience.

## Background Art

These days, digital contents data are commonly distributed on-line such as Internet. The digital contents refer to the concept of encompassing all information required in markets or by consumers, such as market search data, on-line education contents, economically useful database, etc. in addition to the conventional literary works, pictures, cinema, music and games.

Such digital contents are generally created or developed with much effort. The authors wish to protect their own works on-line as well as off-line. However, it is a reality that the existing copyright laws or other systems fail to protect the authors' copyrights to a full extent due to the unlimited duplication and distribution of the works that are a nature of on-line.

Recently, digital contents protection technologies are actively being developed to control particular acts of the contents users by using technical methods in addition to the legal measures for the purpose of blocking duplication, distribution and diverse acts that are not available off-line as well as of protecting profits of the authors.

An example is that only the users authenticated through password, etc. can download and use particular digital data.

Use of the digital contents requires a more complicated technique of security these days. To be specific, unauthorized use is not done from the beginning. The user can first download the digital contents upon payment of the fee, and then store the digital contents in a storage device such as hard disk without authorization. In the secondary act of duplicating or distributing the stored file, the user infringes the copyright of the author. Here, a complicated situation occurs such that control should be discerned against the legitimate use in the user's primary act and the illegal use in his/her secondary act.

The measure conceived to resolve such problem is a streaming method. According to the streaming method, data are not fixedly stored in a storage device such as hard disk. Rather, data are stored in the RAM memory of a computer system in a frame or a data block unit when the data are downloaded in real time so as to be used in a moment. Once a frame is completely downloaded, all the data of the frame are deleted. However, this method poses a problem of user inconvenience due to the communication velocity or other compression, e.g., failure to smoothly streaming the screen and frequent data congestion in case of an active visual image.

Under the circumstances, it is mandatory to invent a method of allowing a user to download and execute an entire data file while blocking the user from storing and using the file without authorization.

A suggestion has been made to meet such need by encrypting and distributing an entire data file. To be specific, an encrypted data file can be decrypted by an authenticated key only. Thus, a data file *per se* cannot be executed by an authorized

execution program without a key. This method is one of the most popular methods these days.

However, this method also has a drawback as briefly explained herein below with reference to FIG 1. The most fundamental problem is that, in order to load contents data on an execution program (120), it is necessary to decrypt an entire encrypted data file (100) with a key and temporarily store the decrypted contents 130 in a storage device. At this stage, if a user copies or transmits the decrypted data to outside (140), the encryption fails to perform its original function and becomes void. Because of this problem, suppliers or contents manufacturers become resistant to trust consumers, and contents data distribution is restricted as a consequence in reality.

Digital rights management (DRM), which is a technology of preventing illegal duplication of digital contents, recently draws more attention from the public and is considered as a sole substitute for the above problem at present.

In general, the DRM refers to a series of hardware and software services and technologies for confining use of digital contents to authenticated users only. The major theme and technologies of the DRM are contents encryption technology, watermarking technology for indicating copyrights, technology of usage policy expression for indicating contents use rules and copyrights as well as technology of storing and processing contents use specification and charging information, etc.

Of the above functions of the DRM, the encryption technology is to prevent unauthorized duplication of the secondary act as stated above by designing a contents data execution program to house a DRM controller, which is the only device to decrypt the encrypted contents data that have been distributed and used. FIG 2 is a schematic diagram illustrating this DRM technology. While decryption is critical to execute the

encrypted contents data, unlike the conventional method, a DRM controller (220) is housed within or linked to the execution program. Therefore, decrypted data need not be stored outside of this area but are transmitted to the execution program (210), which is housed or co-exists therein. Thus, unauthorized users are not allowed to access the decrypted contents that are processed within this area. As a consequence, the users cannot duplicate or distribute the contents without authorization.

The problem of this technology is that the methods of encryption and decryption can be infinitely diversified depending on a company developing the DRM controller. Further, since the contents data to be executed by a single execution program would be the data decrypted by a variety of DRM, this execution program is supposed to house all the variety of DRM controllers. This is impractical in reality. Thus, the suggested method is not only a temporary measure but also another obstacle in distributing the contents.

## Disclosure of Invention

It is, therefore, an object of the present invention to provide a method and a system for protecting digital data that are capable of safely inputting/outputting and executing digital data.

It is another object of the present invention to provide a digital protection method and a system therefore capable of safely transmitting digital data to an authenticated predetermined execution program only in the course of executing digital data.

It is still another object of the present invention to provide a file system-based distribution protection system of digital contents data that enables an execution

program to control execution of the digital data regardless of an encryption method of a DRM controller.

It is still another object of the present invention to provide a system and a method for protecting decrypted contents data from unauthorized duplication or distribution while processing digital data supplied in streaming method as well as a program storage medium that realized the system and the method in the form of software.

It is still another object of the present invention to provide digital contents data distribution system, which is convenient for a user and relatively safe for a copyright holder of the contents.

To achieve the above objects, there is provided a security system for execution of digital data according to the present invention, comprising: first storage means for receiving and storing digital data; access control means for storing identification information about a digital data execution program; determination means connected to the access control means for determining whether or not the identification information about a data-requesting execution program has been recorded in the access control means if the data-requesting execution program calls for a digital data file; and transmission means connected to the first storage means and the determination means for transmitting the digital data to the data-requesting execution program from the first storage means so as to be executed upon receipt of a determination signal that the identification information about the execution program has been recorded from the determination means.

The security system according to the present invention may further comprise encryption means connected to an input terminal of the first storage means for

encrypting the digital data and transmitting the same to the first storage means.

The security system according to the present invention may further comprise first decryption means for decrypting the digital data and transmitting the same to the transmission means if the digital data has been stored in the first storage means in an encrypted form.

The security system according to the present invention may further comprise validity determination means connected to the access control means and including predetermined validity conditions preset in association with the digital data for determining whether or not a command for execution of the digital data satisfies the validity conditions upon receipt of the command for execution of the digital data, and transmitting to the access control means the identification information about the digital data execution program so as to be stored only when the command for execution is determined to be valid.

The digital data may be encrypted using a one-time key. In other words, the security system for executing digital data includes filtering means, which comprises identification information about a predetermined execution program for enabling the predetermined execution program to execute the digital data, and comparing the identification information about the predetermined execution program with the identification information about a data-requesting execution program, if the data-requesting execution program calls for the data for executing purpose, so that the digital data may be execute with respect to the call only when the two kinds of identification information accord with each other.

The present invention also provides a method of allowing only a predetermined execution program to execute digital data. The method according to the

present invention comprises the steps of: storing digital data; securing identification information about a predetermined execution program; securing identification information about a data-requesting execution program if the data-requesting execution program calls for the digital data for the purpose of execution; comparing the identification information about the predetermined execution program with the identification information about the data-requesting execution program; and filtering so that the digital data can be executed in response to the call by the data-requesting execution program only when the identification information about the predetermined execution program accords with the identification information about the data-requesting execution program.

The streaming data protection system according to the present invention comprises: control means for supplying information about streaming data requested for execution, and identification information about an execution program capable of executing the streaming data; access control means for storing the supplied identification information about the execution program; filtering means connected to the access control means for determining whether or not the identification information about a data-requesting execution program has been stored in the access control means if the data-requesting execution program requests execution of the streaming data, and in the affirmative, passing the request for execution; and streaming data supply means for requesting streaming data based on the supplied streaming data information upon receipt of the request for streaming data passed through the filtering means, and supplying the streaming data to the execution program that requested the streaming data upon receipt of the requested streaming data.

## Brief Description of Drawings

The above objects, features and advantages of the present invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings, in which:

FIG 1 is a block schematic diagram illustrating the conventional method of using digital data without authorization;

FIG 2 is a block schematic diagram illustrating the conventional method of executing a DRM controller;

FIG 3 is a block schematic diagram illustrating a filter driver system according to the present invention;

FIG 4 is a block schematic diagram illustrating a method of operating the filter driver system according to the present invention;

FIG 5 is a block schematic diagram illustrating an encrypter/decrypter according to a best mode for carrying out the present invention;

FIG 6 is a block schematic diagram illustrating a method of registering a program that can be executed according to another best mode for carrying out the present invention;

FIG 7 is a block schematic diagram illustrating a division of storage areas according to another best mode for carrying out the present invention;

FIG 8 is a block schematic diagram illustrating a streaming data processing according to another best mode for carrying out the present invention;

FIG 9 is a block schematic diagram illustrating an application to a remote storage of streaming data according to another best mode of the present invention;

FIG 10 is a block schematic diagram illustrating a supply of streaming data

information to external streaming data supply means according to another best mode for carrying out the present invention; and

FIG 11 is a block schematic diagram illustrating streaming data buffering means according to another best mode for carrying out the present invention.

## Best Modes for Carrying out the Invention

Best modes for carrying out the present invention will now be described with reference to the accompanying drawings. In the following description, same drawing reference numerals are used for the same elements even in different drawings. The matters defined in the description are nothing but the ones provided to assist in a comprehensive understanding of the invention. Thus, it is apparent that the present invention can be carried out without those defined matters. Also, well-known functions or constructions are not described in detail since they would obscure the invention in unnecessary detail.

FIG 3 is a block schematic diagram illustrating a client system, in which a filter driver system according to the present invention has been realized.

The filter system in FIG 3 comprises a filter driver controller 300 for controlling a filter driver 330 so as to generate a particular area 320 in a storage device 310 of a client system and allow a predetermined execution program only to access the particular area 320, and a filter driver 330 for determining a data calling of a registered predetermined execution program only as being valid while controlling input/output of all the data in the particular area 320.

The following is a detailed description of its operational mechanism. A filter driver controller 300 commands that a particular storage area 320 is distinctively

generated within a storage device 310 of a client system ( step S1). Identification information for identifying the particular storage area 320 is notified to a filter driver 330 (S2). Generation of the particular area may be directly performed as stated above or may take an indirect form of commanding the filter driver 330 (S3) to generate the particular storage area (S4) and to report information about the particular storage area 320 generated by the filter driver 330 to the filter driver controller 300 (S5).

As a next step, the filter driver controller 300 notifies identification information about a selected or predetermined execution program to the filter driver 330 so as to be registered therein (S6). The registered data 340 perform a role of a kind of access control list. An execution program is determined by this list so that data within the particular storage area can be called and execute only by it.

FIG 3 exemplifies cases of registering B.exe as a writable execution program in the particular storage area A 320 and C.exe as a readable execution program, and D.exe as a writable/readable execution program. Also, E.exe is an execution program not registered in the filter driver as for example. In that case, if a data file "abc.txt" is stored in the storage area A 320, this data file is writable by B.exe or D.exe, and could be called and readable by C.exe and D.exe only. Thus, unauthorized execution programs cannot read or store the data. Access to the data itself is blocked against E.exe, for example, which is a program not authorized or registered to read or write into the data.

Such characteristics of the present invention are powerful measures of solving problems of the conventional method of protecting digital contents or of the DRM controller. To be specific, the conventional system had no remedy for blocking a user's leakage or storage of the contents data stored in a memory in a decrypted form even in

a short period of time prior to be executed by an execution program. Further, in the method of housing a DRM controller and an execution program to avoid such occasion, the DRM controller and the execution program should be in a pair. As mentioned beforehand, however, variety kinds of both the DRM controller and the execution programs pose a problem of limitless combination of pairs.

All these problems have been solved by the present invention at once. The present invention prevents calling and storing the decrypted contents data prior to executing without authorization. Since only a registered execution program is allowed to access and execute the digital data while the registered program is not able to store or distribute the data as previously registered, a consumer or an end user becomes unable to perform any acts other than accessing or executing the data with the execution program. For instance, in FIG 3, if the read-only execution program C.exe is registered, a consumer's unauthorized acts such as copying or redistributing are fundamentally blocked. Moreover, the consumer cannot arbitrarily manipulate what kind of an execution program shall be registered. The manipulation is determined by the filter driver controller according to the present invention.

The construction of FIG 3 is not subject to a particular DRM controller. Therefore, no problem arises such that an execution program should house all kinds of DRM controllers, as mentioned with respect to the conventional DRM controller.

The following is a description of a best mode for carrying out the present invention in association with the conventional DRM controller.

In the present invention, the DRM controller performs a function of encrypting digital data inputted to a client system to be inherent to the domain where the DRM controller belongs. Otherwise, if there exists any extension of the

downloaded digital contents, any contents use information transmitted in addition to the corresponding contents data, or any usage policy, the DRM controller commands an operating system to execute a corresponding execution program based thereon. The DRM controller also performs a function of receiving identification information supplied for the corresponding execution program so as to be transmitted to the filter driver of a subordinate file system.

However, the DRM controller does not necessarily perform the aforementioned functions, which may be performed by a module of the operating system *per se* or by other methods. Thus, it would be obvious for those skilled in the art that the best mode described herein is merely to assist in better understanding of the present invention and that the present invention does not necessarily be used together with the DRM controller.

The identification information or a fingerprint of the execution program referred to in the present invention does not only refer to the identifier supplied by an operating system but also may be a code authentication certificate or an execution image, etc. of a predetermined execution program. In short, the identification information about an execution program is a concept of encompassing all the information that can distinguish an application program or an execution program from other applications. In addition, the execution program referred to in the present invention encompasses all the execution programs that are capable of executing digital contents data.

The contents usage policy data transmitted together with the contents data includes all the information that can effectively use the contents. The use information may be determined by an agreement concluded between a user and a contents provider

such that the corresponding contents is available three times only or for one week from the downloaded date for example. The usage policy data may be downloaded together with the contents data or may be renewed by frequently downloading them from a server, etc. of the contents provider.

FIG 4 shows an application of the filter driver system according to the present invention to a DRM controller in general. Referring to FIG 4, the contents data abc.txt as requested from a client system is inputted to an input device (460) of a client system through network such as Internet or by means of a storage medium such as CD-ROM, etc. (S2). The contents data are encrypted by means of a DRM controller 440 in an original encryption method inherent to the domain where the DRM controller belongs. The encrypted data are stored in a storage device 450 of the client system (S3). The data are decrypted later by the DRM controller. The encryption and decryption techniques of the DRM controller are variable depending on the DRM controller supplier, and are well known to those skilled in the art.

Meanwhile, the filter driver controller 400 generates a particular storage area A 420 within a storage device 410 in advance or in each execution.

In order to execute the stored contents data abc.txt in a predetermined execution program B.exe, the consumer should select the corresponding contents by means of a browser or by other means in his/her own client system. If a signal that the contents has been selected is inputted to the DRM controller 440, the DRM controller first checks the usage policy data that is owned by himself/herself or receivable from a remote server so as to determine whether or not this selection and execution is effective.

As described above, this usage policy is to check whether or not the corresponding contents are available. For instance, if the user is allowed to use the

contents three-times only, the validity can be checked by means of a counter and a comparator housed therein because the number of use is counted and stored. If the user is allowed to use the contents for several days, the validity can be checked by means of a system clock. Such checking conducted within the client system only may be particularly referred to a local authentication. Another authenticating method other than the local authentication is to have the DRM controller, which has recognized an execution command, be automatically connected to an authentication server (not shown in the drawing) of the contents provider so that a validity can be checked and authentication can be obtained therefrom. This case is designed that the DRM controller does not have a usage policy and authentication is obtained from the server from the beginning. This is referred to as a remote authentication according to the present invention. It is out of question that, in case of the local authentication as well, a new usage policy may be downloaded and used by connecting to a server if the usage policy has been renewed and the allowable number of use of the contents has been changed.

If the request command by the user is determined to be valid, the contents data stored in the storage device 450 are called (S4) and decrypted by the DRM controller 440. Subsequently, the decrypted data are stored in the particular storage area A 420 via a filter driver 430 (S5).

If notified from the DRM controller 440 that the contents data abc.txt are to be used or executed(S6), the filter driver controller 400 registers identification information about an execution program, for example, B.exe authorized to execute the contents data in a list 470 in the filter driver 430 of the file system(S7).

The filter driver 430 of the file system is a module, which is controlling

input/output of all the data with respect to the particular storage area A 420. Here, the filter driver refers to an interface between the particular storage area and the execution program, which blocks an access to the digital contents data file of an unauthorized execution program by filtering all the file-related system calls. Further, since data are transmitted between the execution program and the filter driver in the decrypted form, no invasion should be made between the two. Therefore, the filter driver always checks the communication path so as to prevent interruption of its own act by any module, file or command between itself and the execution program authorized from a module within the filter driver that is transmitting the decrypted data. If any module or file is found to interrupt its own act, the filter driver also clears the communication path by advancing itself toward the execution program. Also, if any invasion is found, the filter driver automatically reports or removes the invasion.

In general, the aforementioned functions can be realized by means of a function pointer within the filter driver. If an immediately earlier filter driver exists when registering the filter driver, the system provides the newly registered filter driver with an address of the earlier filter driver. By using such nature of the filter driver, a test filter driver is registered at a front end thereof toward an execution program of the filter driver according to the present invention either periodically or immediately prior to transmitting the decrypted data to the execution program in preparation for the occasion where an unidentified commands or filter driver has invaded the front end toward the filter driver according to the present invention, and checks an address provided for the test filter driver. If this address is the filter driver according to the present invention, only the test filter driver exists at the front end of the execution program according to the present invention. Otherwise, it means that an invaded filter

driver exists because the address would be of the invaded filter driver. In that case, filter driver registered previously according to the present invention should be cancelled and registered again. If newly registered, the filter driver according to the present invention is located at the most advanced position toward the execution program.

Otherwise, it is possible to perform an encryption so that the call is directly made from the test filter driver to the filter driver according to the present invention. A variety of other methods may be used in addition. The reason for performing such functions is to enable the file system filter driver 430 according to the present invention to freely control the input/output of all the data to and from the storage area A 420.

Thereafter, if an execution request for the contents data abc.txt by a predetermined execution program, e.g., B.exe. is made (S8), the request always passes through the file system filter driver 430. In that case, the file system filter driver checks whether or not B.exe is an execution program registered by the filter driver controller. Upon recognition of a registered execution program by means of the registration list 470 within thereof, the file system filter driver 430 calls and loads abc.txt in response so that B.exe may execute abc.txt S9.

By contrast, if unregistered C.exe calls abc.txt, the file system filter driver 430 fails to find C.exe from the list 470, and terminates the process after notifying an error message or an unauthorized use with respect to the request S10. An attention need to be paid on the function of the present invention of controlling an access for the purpose of an unauthorized use of the contents data stored in the decrypted form while operating independently from a particular DRM controller.

FIG 5 shows the system according to another best mode for carrying out the present invention.

No functional explanation will be provided for the elements in FIG 5 that are identified by the same reference numerals as in FIG 4 since they perform the same or similar functions. The difference of FIG 4 from FIG 5 lies in that an encrypter/decrypter (500) is added. In other words, the contents data are stored in the storage area A 420 of the lower filter driver 430 of the file system in an encrypted form (S11), while when the effective execution program is called and executed, contents data are transmitted in a decrypted form (S12). The key capable of reading the encrypted data should always be stored inside of the filter driver 430 of the file system according to the present invention.

The reason for storing the contents data in an encrypted form and storing the key at the file system filter driver is to prevent an unauthorized user from arbitrarily removing the file system filter driver and incapacitating the system according to the present invention. That is, if the filter driver 420 of the file system is removed, the stored key for decryption is also removed. Therefore, even if an unauthorized user may access the storage area A by removing the filter driver, the user will be unable to decrypt the encrypted data stored within the filter driver.

Here, an encryption technique called one-time key well known to those skilled in the art may be used for the encryption. Encryption and decryption keys are variable in each use in case of the one-time key encryption. Thus, an unauthorized user is unable to know the encryption and decryption keys based on his/her previous use of the same. Before or after undergoing the step (S12), the encrypter/decrypter encrypts and stores the abc.txt in preparation for future use.

FIG 6 illustrates another best mode for carrying out the present invention. No functional explanation will be provided for the elements in FIG 6 that are identified by

the same reference numerals as in FIG 4 since they perform the same functions. FIG 6 shows a method for registering the contents data and their execution program on an access control list 470. For instance, each of the contents data is matched with an execution program in the storage area A, and registered on the list 470. In FIG 6, abc.txt file is execution by B.exe., while efg.txt file is execution by C.exe (S13). Accordingly, the construction in FIG 6 is useful for the case where there exists a separate usage policy according to each contents data file.

FIG 7 shows another best mode for carrying the present invention. No functional explanation will be provided for the elements in FIG 7 that are identified by the same reference numerals as in FIG 4 since they perform the same functions.

In FIG 7, two storage areas A, B are separately installed. The access control list 470 shows separate registrations of execution programs accessible to each storage area. This case shows a more efficient use of the contents data according to the usage policy by allowing separate storage and use of the contents data according to different usage policies. In this case, C.exe can execute the data in the storage area B 490 alone (S14).

Another advantage of the present invention is that a similar method is applicable to processing of streaming data with the same effect. FIG 8 is a schematic diagram illustrating a construction of a streaming data processing method according to a best mode for carrying out the present invention. The best mode in FIG 8 exemplifies decryption and execution of the contents data locally encrypted and stored in a client system, e.g., in a user's PC, in a streaming method. The construction in FIG 8 comprises filtering means 820 for controlling input and output of all data within a file system area 900 at the top of the file system 900 and determining validity of the

commands for data execution, a authorized execution program 830 for executing a data file, a unauthorized program 840, which is unable to execute a data file due to no registration in the filtering means 820, a file system area 900 controlled by the filtering means 820 within a client system in storing and outputting the data, streaming data supply means 850 within the file system for requesting streaming data to outside within the file system area 900 and allowing the streaming data to pass the filtering means 820 and be transmitted to the authorized execution program 830, external streaming data supply means 860 of the file system for receiving encrypted contents data from storage means 870 and decrypting the same as well as for supplying the decrypted data in response to the request for data from the internal streaming data supply means 850, and a controller 810 for registering a predetermined execution program in the filtering means 820 as an access control list and providing a data file to be executed and information about the external streaming data supply means 860.

In the first place, a user desiring to execute contents data selects a contents data file to be executed by means of a searching tools or an Internet browser (not shown in the drawing). If a command for executing the contents data file is received, the controller 810 checks the usage policy to confirm whether or not the command for execution is valid. The usage policy means a policy used to determine whether or not the command for execution is a valid act based on the prior agreement concluded between the content data user and the contents data supplier. For instance, if a user's ID is received, the controller 810 recognizes the user based on the received ID, and determines whether or not the corresponding user has been authorized to use the contents data based on the usage policy. Such usage policy is checked either through real-time communication between the server of the data supplier and the controller 810

on line or through notification to the controller 810 by a module housed in the user's client system. However, the controller 810 does not necessarily perform this function. The controller 810 may receive information about validity conditions from another module or a network so as to proceed with the next step.

If the command for execution is determined to be valid based on the usage policy, the controller 810 selects an authorized execution program 830, and registers the identifying information about the execution program such as its process ID, etc. in the filtering means 820 (S1). FIG 8 is a schematic diagram showing registration of the authorized execution program A.exe 830 in the filtering means 820 as an execution program of C.avi. This information about registration is used as an access control list to the file system area 900 in the future so as to determine that the command for calling the registered authorized execution program 830 only is valid. Details in this respect are either identical or similar to the description in the aforementioned Korean Patent Application No. 10-2001-0034583.

Before or after taking the step (S1), the controller 810 registers the identification information about the contents data file to be executed as well as the path of the data to locate the position thereof and the information about the external streaming data supply means for supplying the streaming data from outside of the file system in the file system area 900 (S1'). Examples of registration in the file system area include any cases such as registration within the internal streaming data supply means 850 or registration in a third module (not shown in the drawing) located within the file system area 900 and capable of supplying the above information by being connected to the filtering means 820 and the internal streaming data supply means 850, etc.

Of all the information to be registered, the information about the external streaming data supply means 860 refers to the information as to which external streaming data supply means should be used. When the internal streaming data supply means 850 requests the data, the information about the external streaming data supply means 860 should be supplied to the internal streaming data supply means because the internal streaming data supply means should know where the external streaming data supply means is located and which the data are requested. FIG 8 exemplifies a case of assuming that the external streaming data supply means is C:\B.exe, registering the assumed external streaming data supply means, and notifying such fact to the internal streaming data supply means 850. In this case, the internal streaming data supply means 850 executes the program B.exe in the storage area C of the client system where it is located, and commands that the data be transmitted. Here, "external" means out of the particular area 900, where input/output of the data are not controlled by the filtering means 820 while "internal" means the particular area 900, in which input/output of the data are restricted and controlled by the filtering means 820.

A concept of URL may also be included in the information about the external streaming data supply means. For instance, the command www.m.com/B.exe signifies that the internal streaming data supply means 850 should access the site www.m.com through an Internal connection tool (not shown in the drawing) and transmit the streaming data by means of B.exe existing in that area. Other external streaming data supply means would be dynamic link library (DLL), etc. that will be described later in further detail.

The information to be registered in the internal streaming data supply means 900 should include the information about the contents data to be executed. Such

information is used to notify to the external streaming data supply means 960 which data the internal streaming data supply means (950) requests. Name of the file and its path may be included in that information. FIG. 8 exemplifies a case of supplying "C:\work\C.avi" as information about the contents data stored in the folder "work" of the storage area "C" in the client system.

When the information about the contents data is supplied, the information about the authorized execution program 830 registered in the filtering means 820 should be supplied in a matched state so that the contents data can be executed. The supply of the information about the authorized execution program 830 may be supplied either through registration in the filtering means as a batch process (S1) or in a separate method (S1'). The reason for matching the information about the authorized execution program 830 with the information about the contents data is because the internal streaming data supply means 850 is able to discern which data should be supplied in response to the call from the authorized execution program 830 through the matched information.

As described above, the internal streaming data supply means 850 performs the function of requesting the external streaming data supply means 860 for particular contents data within the file system area 900 (S3), and receiving the data transmitted in response thereto (S6), and transmitting the data to the predetermined authorized execution program 830 through the filtering means 820 (S7) so that the contents data can be executed. Accordingly, the internal streaming data supply means 850 should have prior knowledge as to which external streaming data supply means to communicate with, as well as the information about the requesting contents data. Such knowledge is supplied in advance by the controller 810 in the step (S1'), as described

above.

The contents data stored in the storage means 870 are encrypted by a particular DRM method. Encryption/decryption by DRM methods is a technique well known to those skilled the art. Each DRM method adopts original method of encryption or decryption. Therefore, the data encrypted in a particular DRM method cannot be decrypted by a DRM decrypter of a different method. The data C.avi in FIG 8 was encrypted by an encrypter of a particular DRM method. Thus, the data can be decrypted by a DRM decrypter of the same method. According to the present invention, the external streaming data supply means 860 performs a decryption function and a mediating function. Therefore, the controller 810 should select the streaming data supply means having the same decryption method as the pre-defined encryption method for the contents data and assign it as the external streaming data supply means, and notify it to the internal streaming data supply means 850.

The external streaming data supply means B.exe 860 requested to transmit data by the internal streaming data supply means searches an area 870 storing the contents data by using the information about the contents data supplied from the internal streaming data supply means 850, and receives the contents data from the searched area.

The external streaming data supply means 860 decrypts the received contents data in the aforementioned method, and transmits the same to the internal streaming data supply means 850. Upon receipt, the internal streaming data supply means 850 discerns which execution program has called the received contents data, and transmits the contents data to the authorized execution program 830 through the filtering means 820. The authorized execution program 830 executes a predetermined amount of

received streaming data, thereby completing the call and execution of the contents data.

The contents data here are streaming data as a part of the entire contents data file. The authorized execution program 830 or the internal streaming data supply means 850 determines an amount of data to be requested at once, and requests the data in a method as described above so as to be executed by the authorized execution program.

In FIG 8, if an unauthorized execution program D.exe 840 calls for a data file to execute the contents data (S8), such call is determined to be invalid by the filtering means because D.exe was not registered in the filtering means 820 in advance. Therefore, error messages appear, and the call for the contents data is not executed.

The system in FIG 8 is characterized by a streaming-type execution of data. That is, decrypted data are stored in a file system area controlled by particular filtering means, and a predetermined execution program only execute the data so as to prevent unauthorized duplication and distribution of the data by a user.

FIG 9 exemplifies another best mode for carrying out the present invention. The external streaming data supply means or the contents data storage means are located outside of the client system but linked thereto through network.

In FIG 9, the external streaming data supply means is www.k.net/B.exe. The location of the data on the network is www.e.net/f.avi. If the data are inputted into the file system area 900 by means of the controller 810, the internal streaming data supply means 850 activates the external streaming data supply means B.exe 960, which is located at www.k.net, by using this information, and commands to call in the contents data file named f.avi from www.e.net. Of course, it is out of question that such command for an execution program or a data file located at another system through

network is execution only upon prior permission. For instance, another server system may permit execution of such command only after checking whether or not such a command has been promised in advance or is valid under an agreement based on a user's ID, which has been inputted by the internal streaming data supply means 850 or the controller 810 in advance. The modules bearing the same drawing reference numerals as in FIG 8 perform the same functions as those of the best mode in FIG 8.

FIG 9 assumed a case that both the data and the external streaming data supply means are located outside of the client system. However, it is also applicable to a case that either one of the data or the external streaming data supply means is located outside of the client system. Although FIG 9 identified outside of the system with an Internet URL for the sake of exemplification, Intranet and other network including peripheral storage means are also applicable to this best mode.

FIG 10 illustrates another best mode for carrying out the present invention. FIGs. 8 and 9 exemplify an indirect method, i.e., supplying information about the contents data within the file system area 900 and notifying the information about the contents data to the external streaming data supply means 860 by using the internal streaming data supply means 850. By contrast, FIG 10 exemplifies a direct method, i.e., notifying information about the contents data to the external streaming data supply means 860 by using the controller 810 (S1"). The external streaming data supply means 860, which has received the information, automatically searches the contents data, and supplies the same to the internal streaming data supply means 850. Here, the external streaming data supply means should be an execution program that can control its own activities.

Passively functioning modules that are controlled by the internal streaming

data supply means may also be the external streaming data supply means. DLL is an outstanding example. In that case, execution and control of the DLL, which is external streaming data supply means, is managed by the internal streaming data supply means 850. Accordingly, the information about the contents data should first be supplied to the internal streaming data supply means. Here, the external streaming data supply means is merely a module that is controlled by the internal streaming data supply means in inputting and transmitting the data. In other words, the external streaming data supply means does not perform an active function such as automatically searching the contents data file and requests transmission of the same.

FIG 11 illustrates another best mode for carrying out the present invention. FIG 11 shows a case of adding a buffer memory 1100 between the internal streaming data supply means 850 and the authorized execution program 830 within the file system 900. Description of the other modules in FIG 11 is omitted here because they are the same as those in FIG 1.

In FIG 11, streaming data are transmitted to the internal streaming data supply means 850 from the external streaming data supply means (not shown in the drawing) (S6). The transmitted data are stored in a buffer memory 1100 by the internal streaming data supply means 850 (S42). The stored data are transmitted to the authorized execution program in response to the call from the authorized execution program 830 (S7'). In the meantime, the successive streaming data are supplied to the buffer memory 1100 by undergoing the same process. An advantage of this best mode lies in a sufficient storage of data that are ready to be transmitted in response to a call for execution by the authorized execution program 830 within the buffer memory, which is a temporary memory pool. This is in preparation for a disconnected execution of

data due to a transmission speed problem caused during execution.

In FIG 11, the authorized execution program first calls for the buffer memory 1100 to search existence of any data (S2'). If no data is searched, the authorized execution program notifies the internal streaming data supply means 850 of such fact to request transmission of data (S41). Subsequently, the internal streaming data supply means communicates with the external streaming data supply means to receive data, and fills the empty space of the buffer memory with the data. Determination for existence of data within the buffer memory 1100 may be performed by the authorized execution program 830 or by the buffer memory 1100 *per se*. Further, the request for data by the authorized execution program (S2') may be performed on an independent basis apart from the transmission of the data by the internal streaming data supply means 850 to the buffer memory 1100. In other words, this method is to fill in the buffer memory 1100 by requesting data from outside of the system without checking a command for request of data by the authorized execution program 830 once the internal streaming data supply means 850 senses an empty state of the buffer memory.

## Industrial Applicability

The advantage of the file system filter driver and the access control list according to the present invention will become more apparent in unauthenticated use.

When a user is to execute the contents data, the contents data are executed in a decrypted form. In that case, the user may attempt to store the decrypted data in his/her own storage means or transmit the data through network.

Such problem is resolved by the system according to the present invention. To be specific, the decrypted data are supplied to an authenticated authorized execution

program only through the file system filter driver. Moreover, the filter driver controller is able to control the authorized execution program which has no function of storage or transmission by itself, or by disabling the authorized execution program's function of storage or transmission on a temporary basis. For these functions, the filter driver controller may control the authorized execution program by blocking transmission of its commands to the driver of the storage means or to the driver of the transmission means.

The file system filter driver determines any command from an unauthorized execution program to execute the contents data as an invalid command based on the access control list so as to prevent execution of the contents data. Accordingly, the user is not able to access even the decrypted contents data with an unauthorized execution program that have not been registered.

Thus, unlike the conventional system, the present invention enables a contents data provider to prevent copyright infringing acts even after the contents data are transmitted to a user.

Besides, the present invention does not require matching of an execution program with decryption methods, unlike the conventional DRM method. Therefore, the present invention can make use of the advantages of the conventional DRM method while effectively reducing the load laid on the system.

According to the present invention, contents data can be used widely among the public because the present invention drastically reduces the risk of the contents data provider.

The execution of data by a streaming method according to the present invention includes all the cases of executing data and call for data for the purpose of

execution, either simultaneously or in order. The contents data according to the present invention refers to all the material data corresponding to the authorized execution program including document files, graphic files, audio files and video files.

The module according to the present invention refers to the most basic unit of performing each function or a complex unit of performing multiple functions of the present invention as described above. Further, the data communication between each module may be performed in a method of exchanging a passive role with an active role between the modules, e.g., request for data or transmission of data, if necessary.

The data communication according to the present invention includes the request for data and transmission of data between the related modules.

The present invention may be produced into a computer program. The produced program may be stored in a recording medium or transmitted by a transmitting medium.

The present invention is not limited to a particular operating system but is applicable to other versions of Windows operating system or Unix and other operating systems. The filtering system according to the present invention is applicable between all general execution programs and data, and not limited to the contents data only. Therefore, while the invention has been shown and described with reference to certain best modes for carrying out the invention, it will be understood by those skilled in the art that various changes in form and details may be made in the present invention without departing from the spirit and scope of the invention as defined by the appended claims.

**What Is Claimed Is:**

1. A data security system for execution of data, comprising:

first storage means for receiving and storing data;

access control means for storing identification information about a predetermined execution program that is authorized to execute the data;

determination means connected to the access control means for determining whether or not the identification information about an data-requesting execution program that requests for the data has been recorded in the access control means; and

transmission means connected to the first storage means and the determination means for transmitting the data to the data-requesting execution program from the first storage means so as to be executed upon receipt of a determination signal that the identification information about the data-requesting execution program has been recorded from the determination means.

2. The data security system of Claim 1, further comprising encryption means connected to an input terminal of the first storage means for encrypting the data and transmitting the same to the first storage means.

3. The data security system of Claim 1, wherein the determination means comprises first decryption means for decrypting the data and transmitting the same to the transmission means if the data has been stored in the first storage means in an encrypted form.

4. The data security system of any one of Claims 1 to 3, further

comprising:

second storage means for storing encrypted data; and

second decryption means connected between the second storage means and the first storage means for decrypting the encrypted data stored in the second storage means and transmitting the same to the first storage means.

5. The data security system of Claim 2, further comprising:

second storage means for storing encrypted data; and

second decryption means connected between the second storage means and the encryption means for decrypting the encrypted data stored in the second storage means and transmitting the same to the encryption means.

6. The data security system of Claim 1, further comprising validity determination means connected to the access control means and having predetermined validity condition(s) preset in association with the data for determining whether or not a command for executing the data satisfies the predetermined validity condition(s) upon receipt of the command for executing the data, and transmitting to the access control means the identification information about the predetermined execution program so as to be stored only when the command for execution is determined to be valid.

7. The data security system of Claim 2 or 3, wherein the data are encrypted by using a one-time key.

8. A data security system for execution of data, comprising filtering

means having stored identification information about a predetermined execution program for enabling the predetermined execution program to execute data, and for executing the data in response to a request for the data made by an data-requesting execution program only when the stored identification information about the predetermined execution program accords with the identification information about the data-requesting execution program upon comparison.

9. The data security system of Claim 8, further comprising storage means connected to the filtering means for storing the data.

10. The data security system of Claim 9, wherein the filtering means comprises decryption means for decrypting the encrypted data and stored in the storage means.

11. The data security system of Claim 9, wherein the filtering means comprises encryption means for encrypting the data and transmitting the same to the storage means when storing the data, and decrypting the data when reading the data from the storage means.

12. The data security system of Claim 1, wherein the transmission means is located at forefront in the direction of the data-requesting execution program.

13. The data security system of Claim 8, wherein the filtering means is located at forefront in the direction of the data-requesting execution program.

14. A method for execution of data by a predetermined execution program, the method comprising the steps of:

storing data;

providing identification information about the predetermined execution program;

providing identification information about a data-requesting execution program upon a request for the data by the data-requesting execution program;

comparing the provided identification information about the predetermined execution program with the identification information about the data-requesting execution program; and

filtering for executing the data in response to the request only when the identification information about the predetermined execution program accords with the identification information about the data-requesting execution program.

15. The method of Claim 14, wherein the step of storing data comprises encrypting and storing data.

16. The method of Claim 15, further comprising the step of decrypting the encrypted data.

17. The method of Claim 16, wherein the step of decrypting the data is performed when the step of filtering is performed.

18. A medium for transmitting computer-readable programs, comprising filtering means having stored identification information about a predetermined execution program to execute data by the predetermined execution program only, and for executing the data in response to a request for the data made by a data-requesting execution program only when the identification information about the predetermined execution program accords with the identification information about the data-requesting execution program upon comparison.

19. The medium of Claim 18, wherein the filtering means comprises encryption means for encrypting the data when storing the same, and decrypting the data when reading the same.

20. A medium for storing computer-readable programs, comprising filtering means having stored identification information about a predetermined execution program to execute data by the predetermined execution program only, and for executing the data in response to a request for the data made by a data-requesting execution program only when the identification information about the predetermined execution program accords with the identification information about the data-requesting execution program upon comparison.

21. The medium of Claim 20, wherein the filtering means comprises encryption means for encrypting the data when storing the same, and decrypting the data when reading the same.

22. The data security system of any one of Claims 1 to 7, wherein the data are digital contents data.

23. A streaming data protection system for protecting streaming data executed by a streaming method, the system comprising:

control means for supplying information about streaming data requested for execution and identification information about an execution program capable of executing the streaming data;

access control means for storing the supplied identification information about the execution program;

filtering means connected to the access control means for determining whether or not the identification information about a data-requesting execution program has been stored in the access control means upon receipt of a request for execution of the streaming data by the data-requesting execution program, and in the affirmative, passing the request for execution as a valid request; and

streaming data supply means for requesting the streaming data based on the supplied streaming data information upon receipt of the request for streaming data passed through the filtering means, and supplying the streaming data to the execution program that requested the streaming data upon receipt of the requested streaming data.

24. The streaming data protection system of Claim 23, further comprising streaming data storage means for storing the requested streaming data, and supplying the stored streaming data to the streaming data supply means in response to the request from the streaming data supply means.

25. The streaming data protection system of Claim 23, further comprising decryption means for decrypting the requested streaming data, if encrypted, and supplying the same to the streaming data supply means.

26. The streaming data protection system of Claim 24, wherein the streaming data supply means comprises:

first streaming data supply means for performing data communication with the execution program by passing through the filtering means only; and

second streaming data supply means for performing data communication with the first streaming data supply means not necessarily passing through the filtering means upon receipt of the streaming data from the streaming data storage means.

27. The streaming data protection system of Claim 26, wherein the second streaming data supply means may perform its function on an independent basis, and the information about the streaming data requested for execution is supplied to the second streaming data supply means by the control means.

28. The streaming data protection system of Claim 24, wherein the streaming data storage means is remotely located from the streaming data supply means but is linked thereto through network.

29. The streaming data protection system of Claim 26, wherein the second streaming data supply means is remotely located from the first streaming data

supply means but is linked thereto through network.

30. The streaming data protection system of Claim 23, further comprising buffering means between the execution program and the streaming data supply means for supplying the streaming data to the execution program through the buffering means upon receipt of the requested streaming data.

31. A method for protecting streaming data executed by a streaming method, comprising the steps of:

supplying information about streaming data requested for execution and identification information about an execution program capable of executing the streaming data;

storing the supplied identification information about the execution program in an access control means;

determining whether or not the identification information about a data-requesting execution program has been stored in the access control means if the data-requesting execution program requests execution of the streaming data, and in the affirmative, transmitting to a steam data supply means the request for execution as a valid request;

requesting the streaming data by the streaming data supply means upon receipt of the request for streaming data based on the supplied information about the streaming data; and

supplying the streaming data to the execution program that requested the streaming data upon receipt of the streaming data by the streaming data supply means.

32. The method of Claim 31, further comprising the step of storing the streaming data for the purpose of supplying the streaming data to the streaming data supply means in response to the request therefrom.

33. The method of Claim 31, further comprising the step of decrypting the requested streaming data, if encrypted, and supplying the same to the streaming data supply means.

34. The method of Claim 31, further comprising the step of buffering the streaming data prior to supplying the requested streaming data to the execution program upon receipt of the same by the streaming data supply means.

35. A computer program storage medium for protecting streaming data executed by a streaming method, comprising:

control means for supplying information about streaming data requested for execution and identification information about an execution program capable of executing the streaming data;

access control means for storing the supplied identification information about the execution program;

filtering means connected to the access control means for determining whether or not the identification information about a data-requesting execution program has been stored in the access control means upon receipt of a request for execution of the streaming data by the data-requesting execution program, and in the affirmative,

passing the request for execution as a valid request; and

streaming data supply means for requesting the streaming data based on the supplied streaming data information upon receipt of the request for streaming data passed through the filtering means, and supplying the streaming data to the execution program that requested the streaming data upon receipt of the requested streaming data.

36. The computer program storage medium of Claim 35, further comprising buffering means located between the execution program and the streaming data supply means for buffering the streaming data prior to supplying the streaming data to the execution program by the streaming data supply means.

FIG. 1

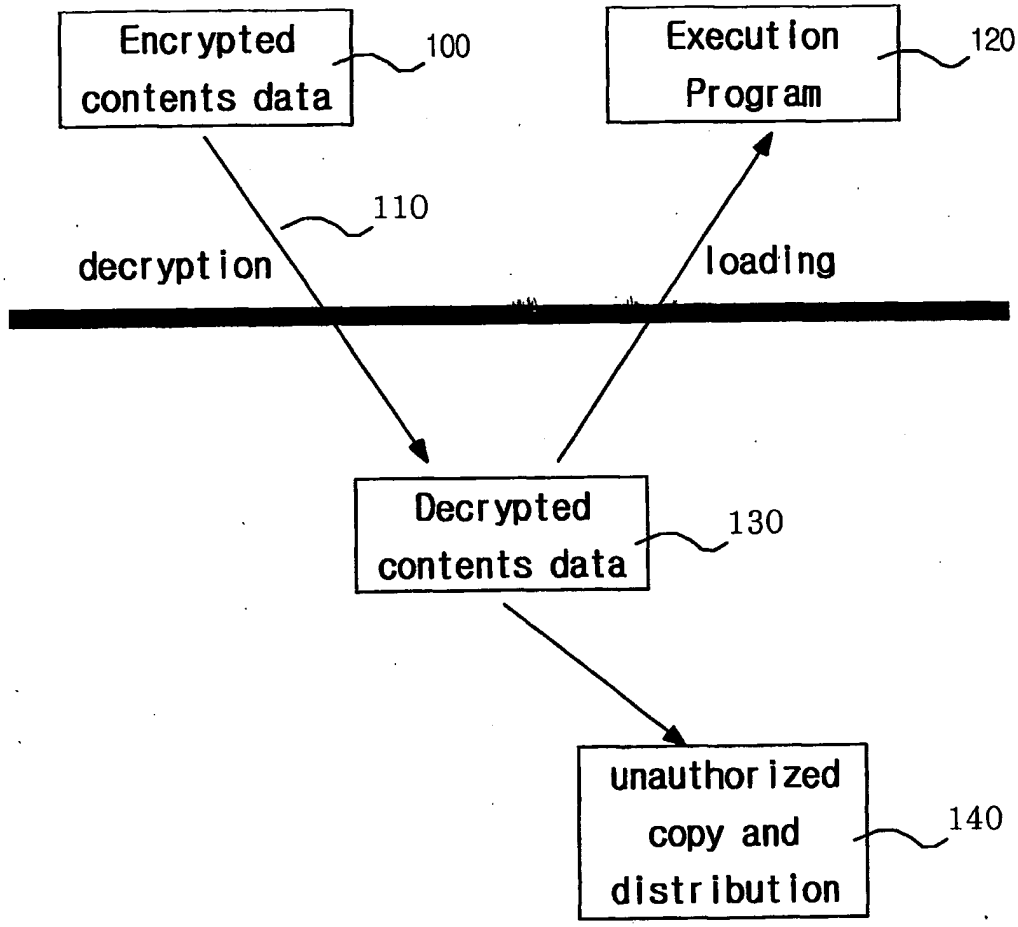

Encrypted contents data
100
Execution Program
120
110
decryption
loading
Decrypted contents data
130
unauthorized copy and distribution
140

FIG.2

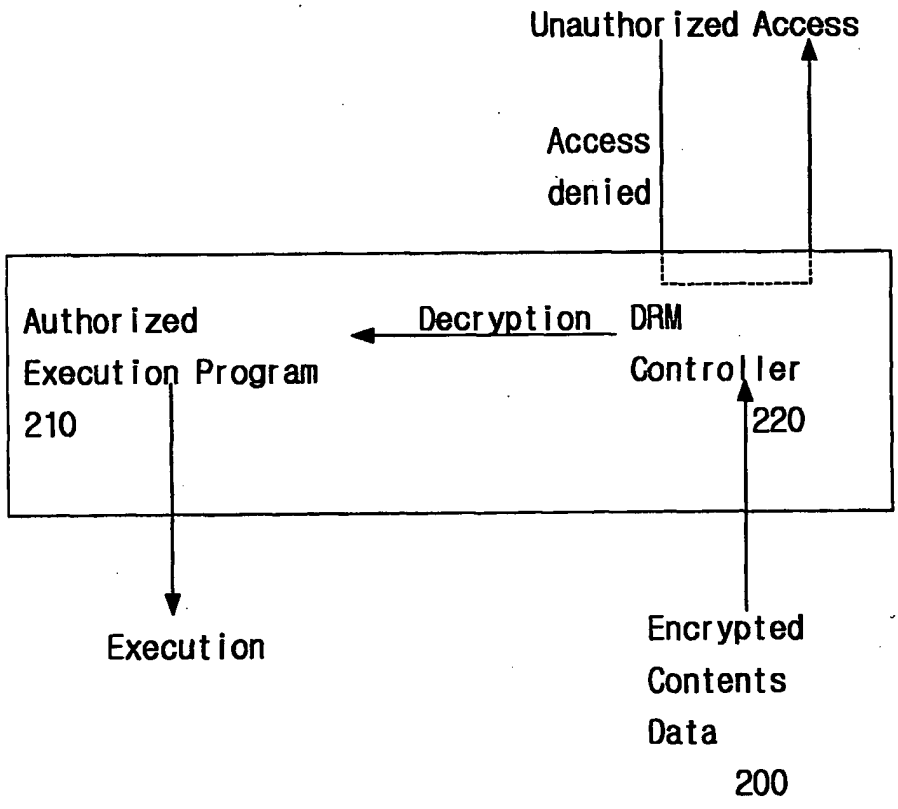

Unauthorized Access
Access
denied
Authorized
Execution Program
210
Decryption
DRM
Controller
220
Execution
Encrypted
Contents
Data
200

**FIG.3**

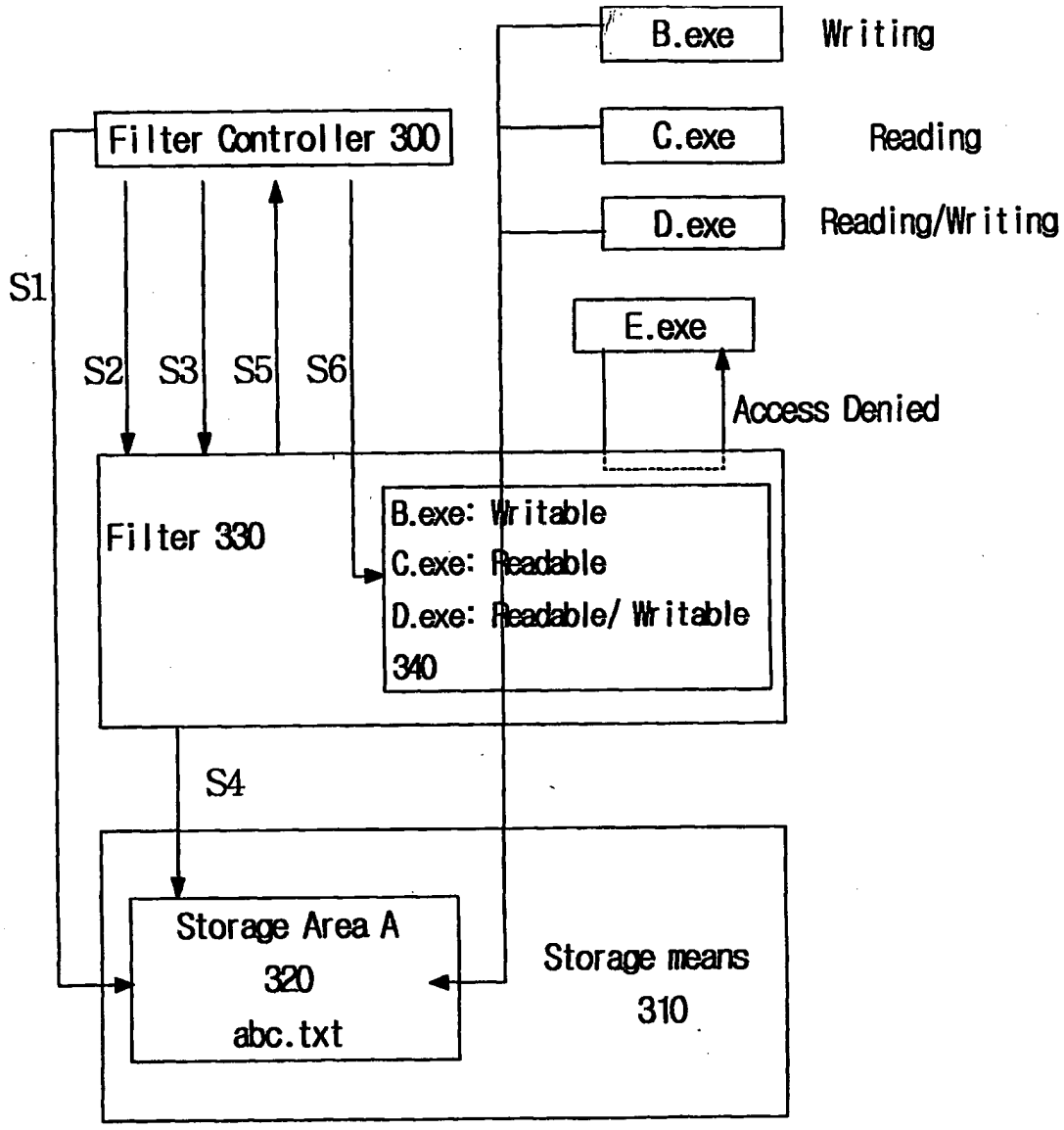

B.exe
Writing
Filter Controller 300
C.exe
Reading
D.exe
Reading/Writing
S1
E.exe
S2
S3
S5
S6
Access Denied
Filter 330
B.exe: Writable
C.exe: Readable
D.exe: Readable/ Writable
340
S4
Storage Area A
320
abc.txt
Storage means
310

**FIG.4**

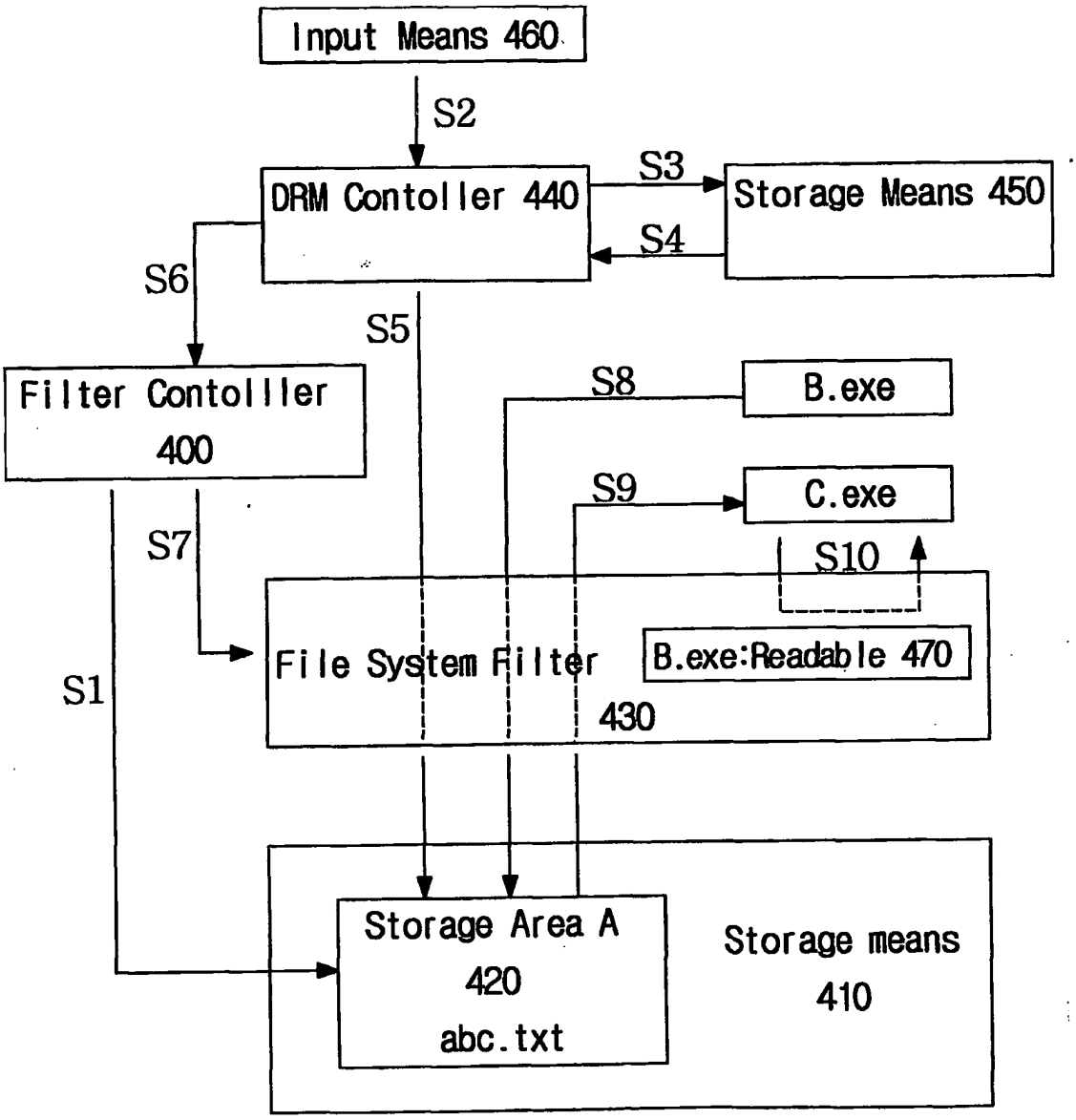

Input Means 460
S2
DRM Contoller 440
S3
S4
Storage Means 450
S6
S5
Filter Contolller
400
S8
B.exe
S9
C.exe
S10
S7
File System Filter
B.exe:Readable 470
430
S1
Storage Area A
420
abc.txt
Storage means
410

FIG.5

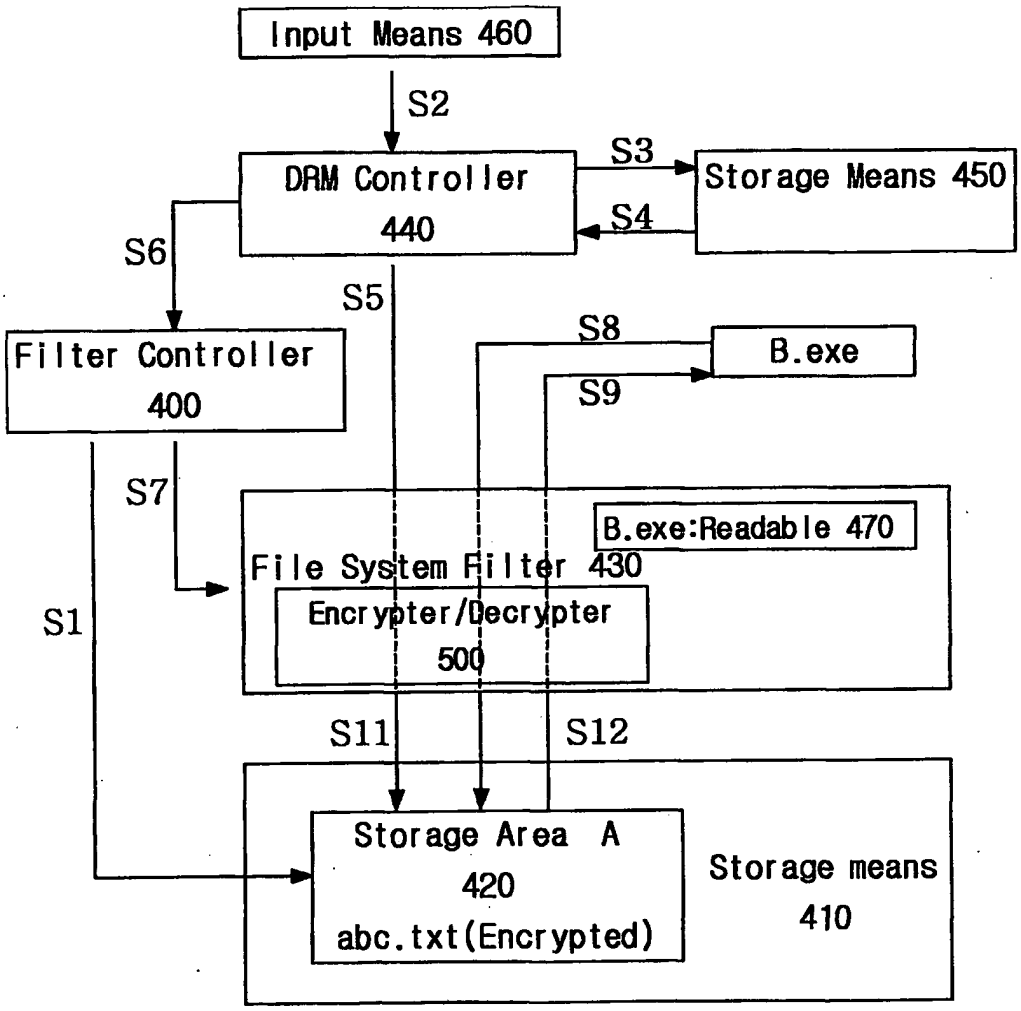

Input Means 460
S2
DRM Controller 440
S3
S4
Storage Means 450
S6
S5
Filter Controller 400
S8
B.exe
S9
S7
B.exe:Readable 470
File System Filter 430
Encrypter/Decrypter 500
S1
S11
S12
Storage Area A 420
abc.txt(Encrypted)
Storage means 410

**FIG.6**

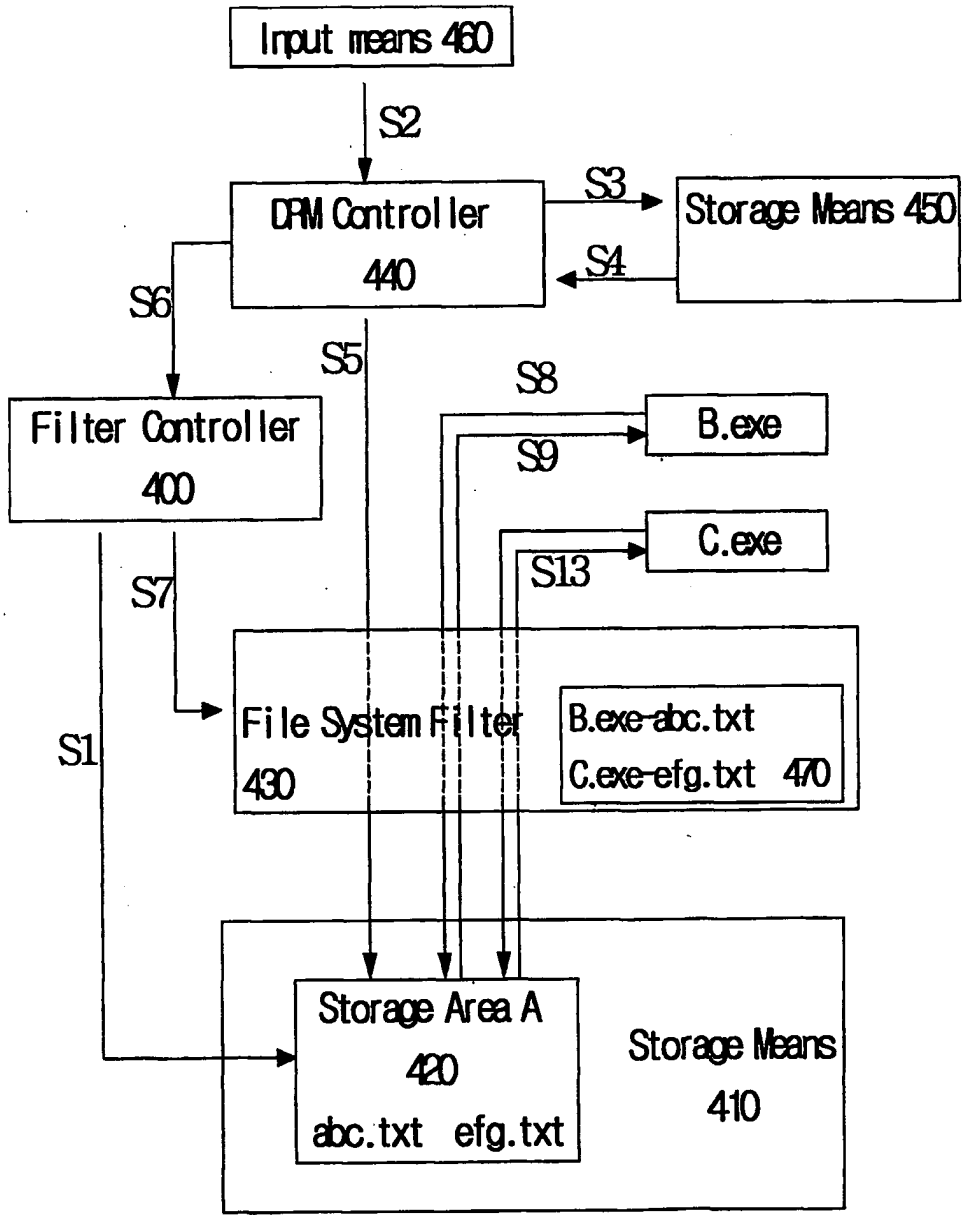

Input means 460
S2
DRM Controller
440
S3
Storage Means 450
S4
S6
S5
S8
Filter Controller
400
B.exe
S9
C.exe
S13
S7
File System Filter
430
B.exe-abc.txt
C.exe-efg.txt 470
S1
Storage Area A
420
abc.txt efg.txt
Storage Means
410

FIG.7

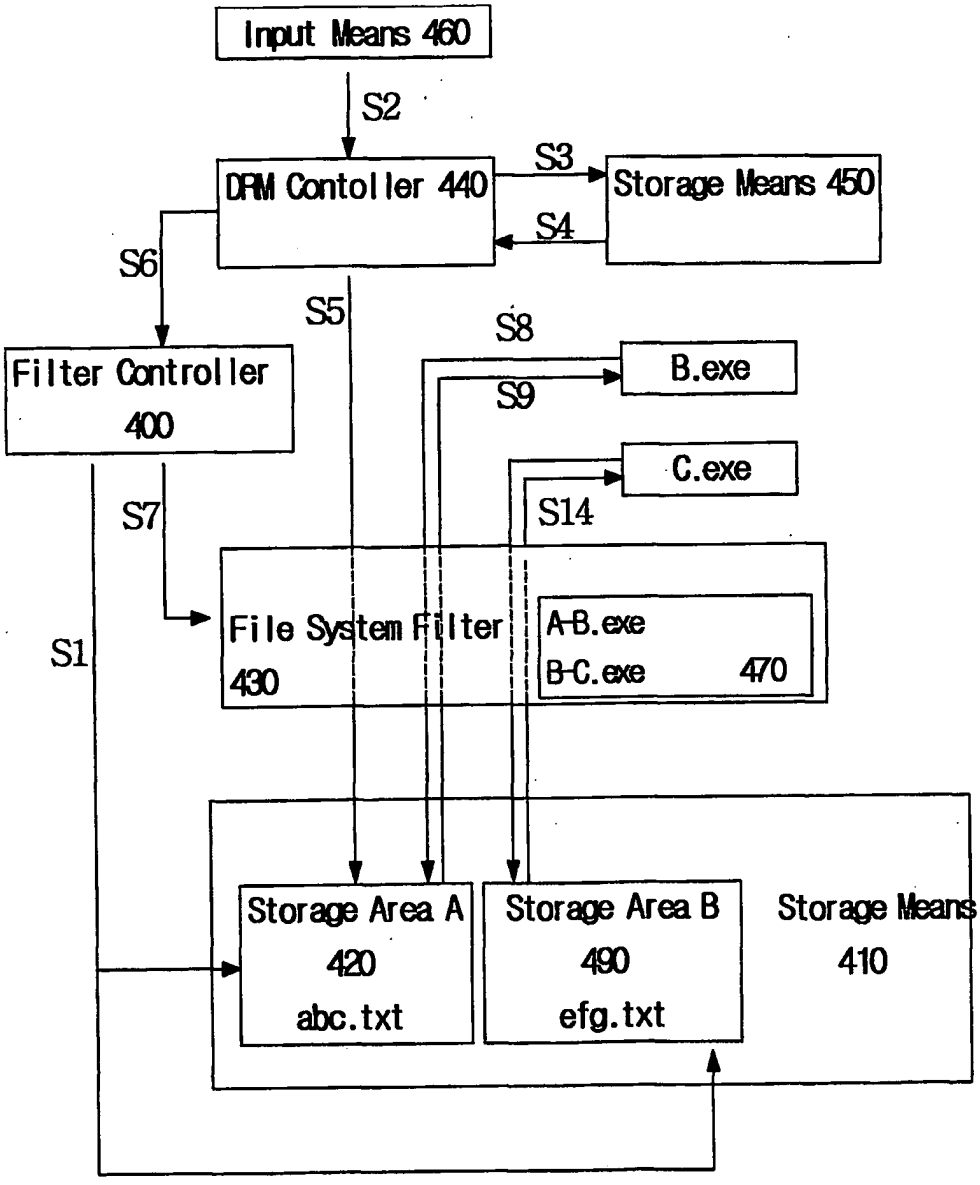

Input Means 460
S2
DRM Contoller 440
S3
Storage Means 450
S4
S6
S5
S8
Filter Controller
400
B.exe
S9
C.exe
S14
S7
File System Filter
430
A-B.exe
B-C.exe
470
S1
Storage Area A
420
abc.txt
Storage Area B
490
efg.txt
Storage Means
410

FIG.8

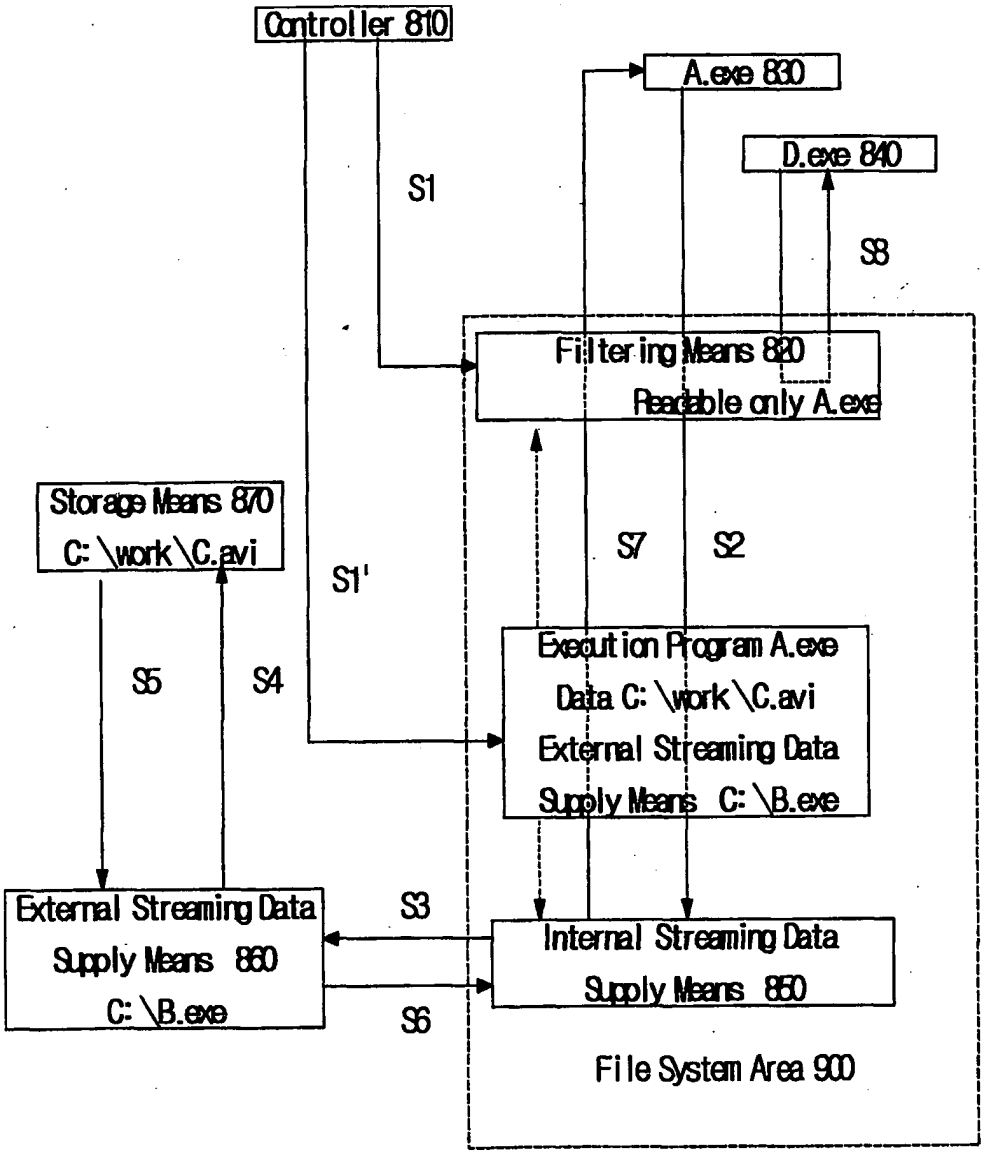

Controller 810
A.exe 830
D.exe 840
S1
S8
Filtering Means 820
Readable only A.exe
Storage Means 870
C:\work\C.avi
S1'
S7
S2
S5
S4
Execution Program A.exe
Data C:\work\C.avi
External Streaming Data
Supply Means C:\B.exe
External Streaming Data
Supply Means 860
C:\B.exe
S3
S6
Internal Streaming Data
Supply Means 850
File System Area 900

**FIG.9**

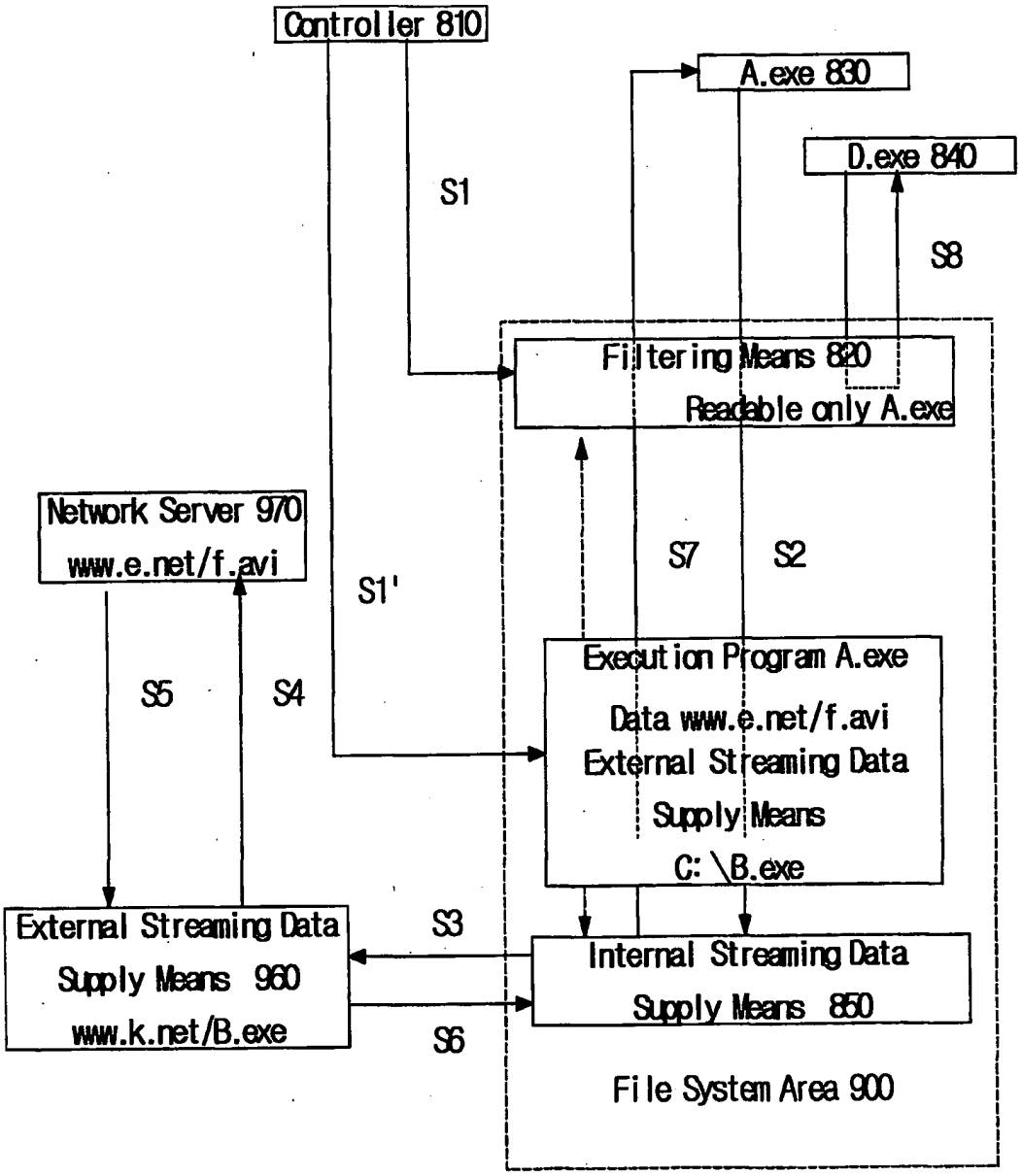

Controller 810
A.exe 830
D.exe 840
S1
S8
Filtering Means 820
Readable only A.exe
Network Server 970
www.e.net/f.avi
S1'
S7
S2
S5
S4
Execution Program A.exe
Data www.e.net/f.avi
External Streaming Data
Supply Means
C:\B.exe
External Streaming Data
Supply Means 960
www.k.net/B.exe
S3
S6
Internal Streaming Data
Supply Means 850
File System Area 900

FIG.10

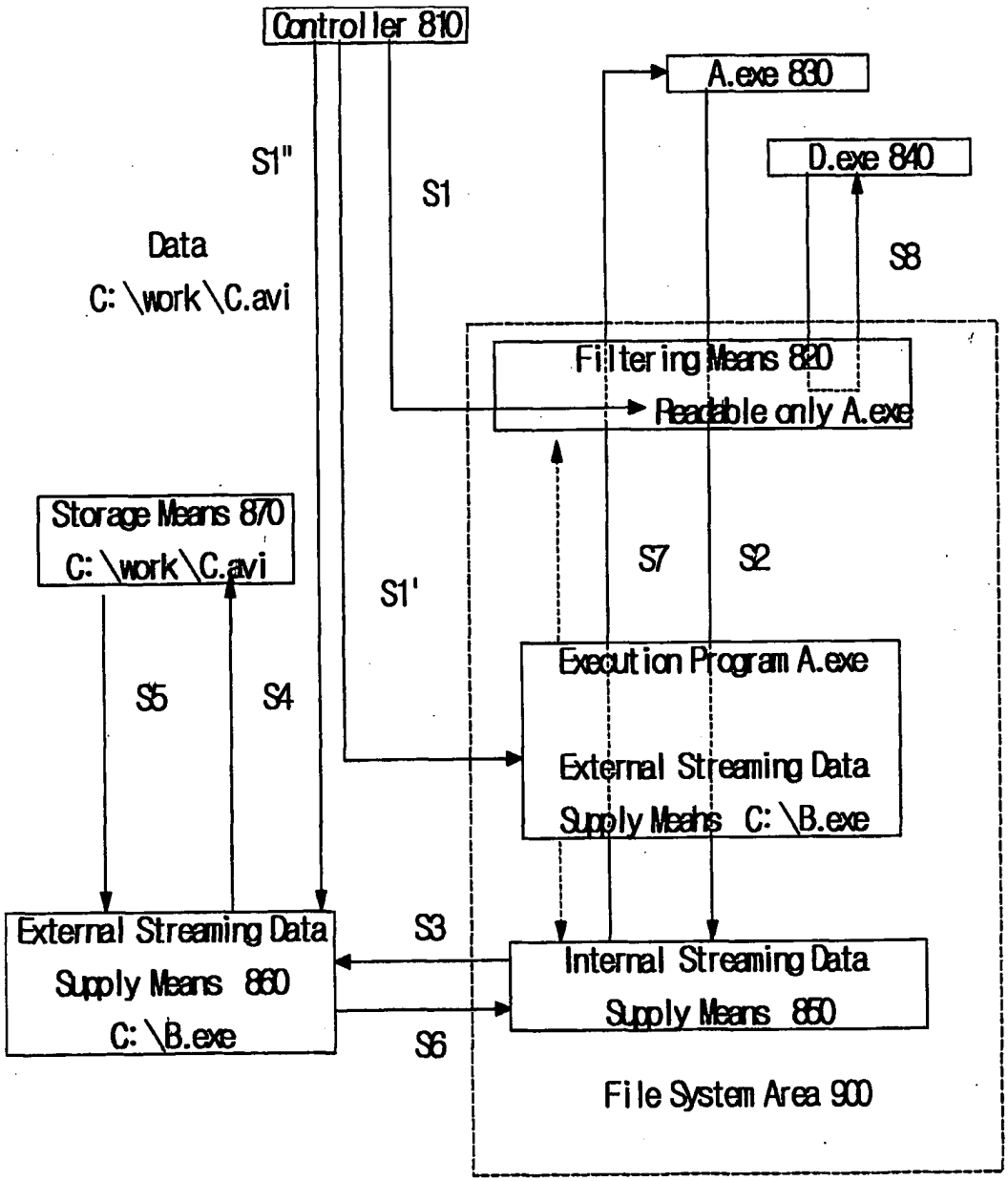
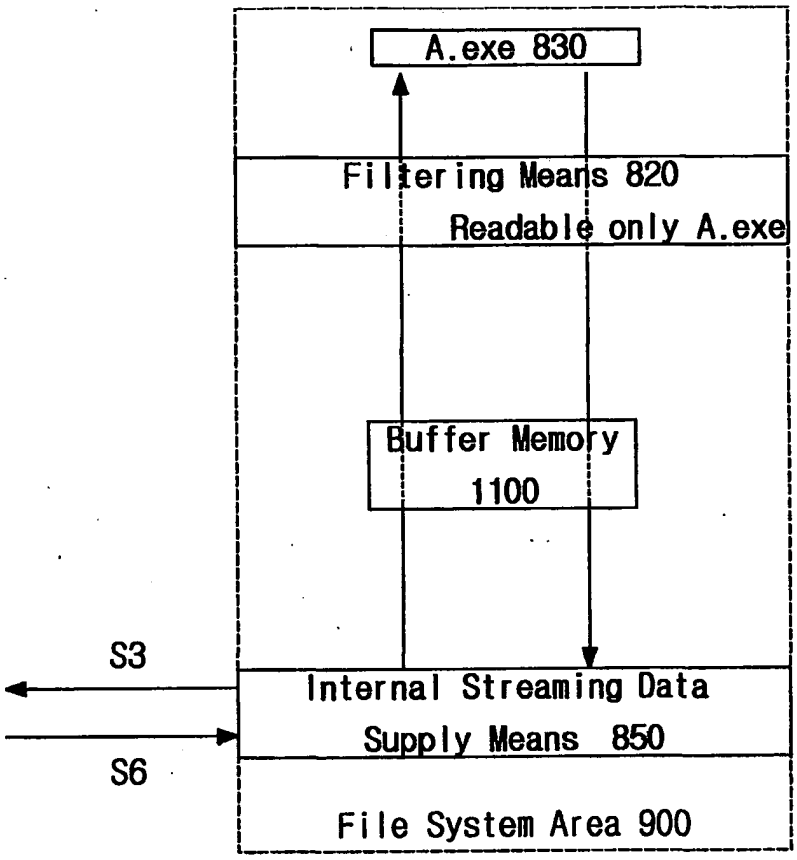

Controller 810
A.exe 830
D.exe 840
S1"
S1
Data
C:\work\C.avi
S8
Filtering Means 820
Readable only A.exe
Storage Means 870
C:\work\C.avi
S7
S2
S1'
Execution Program A.exe
External Streaming Data
Supply Means C:\B.exe
S5
S4
External Streaming Data
Supply Means 860
C:\B.exe
S3
Internal Streaming Data
Supply Means 850
S6
File System Area 900

FIG.11


A.exe 830
Filtering Means 820
Readable only A.exe
Buffer Memory
1100
S3
S6
Internal Streaming Data
Supply Means 850
File System Area 900

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR02/01157

**A. CLASSIFICATION OF SUBJECT MATTER**

**IPC7 G06F 15/00**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC7 H04L 9/00, 9/32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | JP 11-27311 A (CANNON Co.,) 29. JAN. 1999<br>FIG 1, 2, 3, 4, 5, 6, 7, 9-14 ABSTRACT, CLAIMS 1, 2, 3, 4, 5, 7, 9, 11-17 | 1-36 |
| A | US 5,910,987 A (InterTrust Technologies Co.,) 8. JUN. 1999<br>FIG 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 15-31 ABSTRACT, CLAIMS 1, 2 | 1-36 |
| A | US 5,819,089 A (Sterling Software, Inc.,) 6. OCT. 1998<br>FIG 1, 2, 3, 4, 5, 6, 7, 8, 9 ABSTRACT, CLAIMS 1, 2, 3, 4, 5, 6, 7, 8 | 1-36 |
| A | US 6,006,332 A (Case Western Reserve University) 21. DEC. 1999<br>FIG 1, 2, 3, 4, 5, 6, 7, 8, 10-15 ABSTRACT, CLAIMS 1, 2, 3, 4, 5, 6, 7, 11-19 | 1-36 |

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevence

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevence; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevence; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 06 SEPTEMBER 2002 (06.09.2002) | 06 SEPTEMBER 2002 (06.09.2002) |

| Name and mailing address of the ISA/KR | Authorized officer |
|---|---|
| Korean Intellectual Property Office<br>920 Dunsan-dong, Seo-gu, Daejeon 302-701,<br>Republic of Korea<br>Facsimile No. 82-42-472-7140 | LEE, Un Cheol<br><br>Telephone No. 82-42-481-5709 |

Form PCT/ISA/210 (second sheet) (July 1998)